

УДК 519.6

**ПРИМЕНЕНИЕ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ ДЛЯ ЧИСЛЕННОГО
МОДЕЛИРОВАНИЯ ТЕЧЕНИЯ ВЯЗКОЙ НЕСЖИМАЕМОЙ
ЖИДКОСТИ В ОБЛАСТЯХ СЛОЖНОЙ КОНФИГУРАЦИИ
МЕТОДОМ ПОГРУЖЕННОЙ ГРАНИЦЫ**

Е. В. Мортиков¹

Рассматривается применение графических процессоров для моделирования вязкой несжимаемой жидкости в областях сложной конфигурации. Метод погруженной границы используется для описания криволинейных границ на прямоугольных сетках. Для оценки эффективности реализации численных методов на архитектуре графических процессоров моделируется течение вокруг кругового цилиндра и группы круговых цилиндров. В качестве примера задачи с подвижными границами численно воспроизводится течение вокруг кругового цилиндра, совершающего вынужденные гармонические колебания.

Ключевые слова: уравнения Навье–Стокса, метод погруженной границы, CUDA, графические процессоры.

1. Введение. Развитие графических процессоров в последнее десятилетие вызвало рост интереса к возможности использования графических карт в качестве математических сопроцессоров для проведения расчетов, связанных с численным моделированием. Распространение специализированных сред программирования (CUDA, OpenCL) способствует появлению реализаций математических моделей, в которых перенос вычислений на новую архитектуру связан не с целью визуализации обрабатываемых данных, а с целью ускорения работы программ для расчета моделируемых явлений.

Первые работы, посвященные использованию графических процессоров в задачах вычислительной гидродинамики, связаны с попытками совмещения численных расчетов, описывающих течение, и визуализацию потока непосредственно на видеокартах [28, 29]. Широкое распространение в настоящее время получили методы решеточных уравнений Больцмана для воспроизведения течений в сложных областях в связи с достаточной простотой распараллеливания на различные архитектуры, в том числе на графические процессоры, где достигается значительное ускорение по сравнению с вычислениями на центральном процессоре [30, 31, 33, 34]. Тем не менее, их применение в задачах моделирования сжимаемой жидкости, а также для описания подвижных границ, как и аналогичных методов частиц [32], имеет ряд ограничений.

Существующие алгоритмы, основанные на решении сеточных уравнений Навье–Стокса, не так эффективны при реализации на графических процессорах, что связано с необходимостью решения уравнения Пуассона для выполнения уравнения неразрывности, а также учета криволинейных границ при решении задач в сложных областях. Выбор подходящего метода для решения эллиптических уравнений зависит от класса решаемых задач, численной схемы и типа дискретной сетки, покрывающей область. Для ряда задач возможно применение эффективных многосеточных методов [15–17], которые демонстрируют хорошие результаты при переходе на архитектуру графических процессоров [23, 24]. К другим распространенным методам решения уравнения Пуассона относятся итерационные методы градиентного типа [13], однако их быстродействие сильно зависит от выбора предобусловливателя с учетом особенностей вычислительной архитектуры и программной модели.

При моделировании течений в областях сложной конфигурации применение криволинейных сеток, соответствующих границам области, требует больших вычислительных затрат для генерации сетки, особенно в задачах с движущимися или деформируемыми границами. Иной способ представления криволинейных границ состоит в использовании прямоугольных декартовых сеток, при этом необходимы дополнительные модификации исходной системы уравнений или численной схемы для аппроксимации краевых условий. Такой подход не зависит от конкретной геометрии области и не требует разработки сложных алгоритмов покрытия области криволинейными сетками, что делает его более привлекательным для реализации на параллельных вычислительных системах, в том числе для расчетов на графических процессорах.

¹ Научно-исследовательский вычислительный центр, Московский государственный университет им. М. В. Ломоносова, Ленинские горы, 119991, Москва; мл. науч. сотр., e-mail: Evgeny.Mortikov@gmail.com

К методам, сочетающим криволинейные границы и декартовы сетки в задачах вычислительной гидродинамики, можно отнести: метод ступенчатого представления границы, метод скошенных ячеек [40, 41] и метод погруженной границы [1]. Метод ступенчатого представления границы основан на совмещении узлов сетки и точек криволинейной границы. Для многих задач точность приближения границы ступеньками оказывается недостаточной. В методе скошенных ячеек изменяется форма ячеек, имеющих пересечение с границей, что нарушает однородность вычислительной сетки и приводит к схожим недостаткам, как и у методов, использующих адаптивные криволинейные сетки.

Метод погруженной границы был разработан в 70-е годы XX века для моделирования потока крови вокруг сердечного клапана [2]. Для выполнения краевых условий на криволинейной границе в уравнение движения добавлялась специальная функция силы, значение которой вычислялось из выполнения закона Гука для упругой границы. Применение данного метода и его аналогий для задач с твердыми границами приводит к жестким системам уравнений [5–7]. Позднее для подобных задач и для моделирования турбулентных потоков были разработаны варианты метода погруженной границы, в которых влияние криволинейной границы учитывается при численном решении задачи на основе известных полей скорости и заданных краевых условиях [3, 4, 8, 9, 11, 43, 53, 54]. Важным для точности аппроксимации геометрии области в этих методах являются способы интерполяции/экстраполяции значений в точках криволинейной границы, чем и объясняется большое разнообразие в существующих модификациях данного подхода. Различие в интерполяционных процедурах также необходимо учитывать при проектировании программных реализаций на графических процессорах.

При решении задач с нестационарной геометрией метод погруженной границы имеет очевидные достоинства по сравнению с методами, основанными на покрытии области криволинейными сетками, и он успешно применялся для моделирования многих сложных течений [10]. Однако возникает ряд проблем, связанных с отсутствием гладкости в полях скорости и давления при смещении границы на фиксированной декартовой сетке [10, 12, 44, 45]. Существует ряд методов, таких как метод погруженных интерфейсов [42] и др., которые в связи с отсутствием принятого подхода к классификации способов представления криволинейных границ на прямоугольных сетках также можно отнести к классу методов погруженной границы.

В настоящей статье рассматривается реализация метода погруженной границы для воспроизведения течения вязкой несжимаемой жидкости в областях со сложной геометрией и подвижными границами на графических процессорах с помощью технологии программирования CUDA. Движение вязкой несжимаемой жидкости описывается системой уравнений Навье–Стокса, а для численной аппроксимации уравнений на прямоугольной сетке используется метод конечных разностей. Выполнение краевых условий на криволинейной границе достигается методом погруженной границы, предложенным в работе [11], с изменениями, связанными с выбором разностных приближений дельта-функций для операторов интерполяции. Для решения уравнения Пуассона используется предобусловленный метод сопряженных градиентов с известными модификациями для параллельных архитектур [13, 14, 18, 19]. В качестве предобусловливателя рассматриваются реализации на графических процессорах метода Якоби, симметричного метода Гаусса–Зейделя с переупорядочиванием узлов сетки в шахматном порядке, варианта геометрического многосеточного метода с V-циклом, а также предобусловливатель, основанный на приближении обратной матрицы системы [26]. Наиболее эффективным для метода погруженной границы является применение в качестве предобусловливателя геометрического многосеточного метода при решении эллиптического уравнения. Тем не менее, для других формулировок метода погруженной границы, таких как метод фиктивных ячеек [3, 43, 53, 54], возникают сложности с аппроксимацией матрицы системы на грубых сетках.

Ниже обсуждается реализация данной модели как для отдельной графической карты, так и реализация для кластера, объединяющего несколько устройств, где для организации обменов данными используется библиотека MPI. Для демонстрации возможностей метода погруженной границы и численной модели решается ряд двумерных задач вычислительной гидродинамики. Численно воспроизводится ламинарное течение вокруг одного кругового цилиндра и группы цилиндров, расположенных в шахматном порядке. В качестве задачи с подвижными границами моделируется поток вокруг кругового цилиндра, совершающего вынужденные колебания перпендикулярно к набегающему потоку жидкости. Проводится сравнение с реализацией итерационных методов и метода погруженной границы на центральном процессоре.

2. Численный метод. Рассмотрим уравнения Навье–Стокса для вязкой несжимаемой жидкости в ограниченной области Ω с границей Γ , в которой расположено N_b погруженных областей Ω_b^k , ограниченных

замкнутыми достаточно гладкими границами Γ_b^k , $k = 1, \dots, N_b$:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \nu \nabla^2 \mathbf{u} - \frac{1}{\rho} \nabla p + \mathbf{f}(\mathbf{x}, t), \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

$$B\mathbf{u}(\mathbf{x}, t)|_{\Gamma} = 0, \quad \mathbf{u}(\mathbf{x}, t_0) = \mathbf{u}_0, \tag{3}$$

$$\mathbf{u}(\mathbf{x}, t)|_{\Gamma_b^k} = \mathbf{U}_b^k(t), \quad k = 1, \dots, N_b, \tag{4}$$

где $\mathbf{u}(\mathbf{x}, t)$ — вектор скорости, $p(\mathbf{x}, t)$ — давление; ρ, ν — плотность и кинематическая вязкость соответственно. Условия (3) представляют собой начальные и краевые условия на границе области Ω , вид оператора B зависит от типа известных граничных условий. На погруженных границах Γ_b^k рассматриваются краевые условия Дирихле (4). Уравнения движения (1) и уравнение неразрывности (2) определяют неизвестные скорость и давление во всей области Ω без учета криволинейных границ. Функция $\mathbf{f}(\mathbf{x}, t)$ в уравнении (1) соответствует аппроксимации краевых условий (4), а способ ее определения рассматривается далее при описании метода погруженной границы.

Численное решение системы (1)–(4) будем искать методом конечных разностей на прямоугольной сетке Ω_h с разнесенным способом размещения переменных скорость–давление в ячейке. Для интегрирования уравнений по времени используется метод дробных шагов [35–37], конвективные и диффузионные слагаемые в уравнении движения (1) приближаются явной схемой Адамса–Башфорта второго порядка точности по времени. Систему уравнений Навье–Стокса в дискретном виде можно условно записать следующим образом:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + [\mathbf{u} \cdot \nabla \mathbf{u}]^{n+1/2} = \nu [\nabla^2 \mathbf{u}]^{n+1/2} - \frac{1}{\rho} \nabla p^n + \mathbf{f}^{n+1}(\mathbf{x}, t_{n+1}), \tag{5}$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\nabla \phi^{n+1},$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0,$$

где Δt — шаг по времени, $[\mathbf{u} \cdot \nabla \mathbf{u}]^{n+1/2} = \frac{3}{2} \mathbf{u}^n \cdot \nabla \mathbf{u}^n - \frac{1}{2} \mathbf{u}^{n-1} \cdot \nabla \mathbf{u}^{n-1}$, $[\nabla^2 \mathbf{u}]^{n+1/2} = \frac{3}{2} \nabla^2 \mathbf{u}^n - \frac{1}{2} \nabla^2 \mathbf{u}^{n-1}$, краевые условия (3) и (4) заданы на $(n + 1)$ -м шаге, \mathbf{u}^* — промежуточное дивергентное поле скорости, ϕ^{n+1} — поправка к давлению, определяемая как решение уравнения Пуассона с краевыми условиями Неймана на границе области Ω (для разрешимости конечно-разностного уравнения на части границы определяется условие Дирихле):

$$\Delta \phi^{n+1} = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t}. \tag{6}$$

Тогда значения вектора скорости и давления на новом шаге по времени вычисляются следующим образом:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi^{n+1}, \tag{7}$$

$$p^{n+1} = p^n + \rho \phi^{n+1}.$$

Для дискретизации конвективных слагаемых по пространству используется консервативная кососимметрическая форма второго порядка точности [38], а для диффузионных слагаемых — схема центральных разностей. Матрица системы уравнений (6) является симметричной разреженной матрицей (пятидиагональной или семидиагональной в двумерном или трехмерном случаях соответственно).

Выполнение краевых условий на погруженных границах Γ_b^k , $k = 1, \dots, N_b$, достигается добавлением функции $\mathbf{f}(\mathbf{x}, t)$ в уравнение движения (1). Границы Γ_b^k при решении численной задачи представляются набором из P_b^k точек и заданных краевых условий $\mathbf{u}(\mathbf{x}, t)|_{\Gamma_b^k} = \mathbf{U}_b^k(t)$. Введем оператор интерполяции L поля значений, заданных на сетке, в точки границы и оператор проектирования L^* в узлы сетки по значениям поля на дискретной границе:

$$\mathbf{F}^k = L \mathbf{f}, \quad F_i^k = \sum_{s \in \Omega_h} f_s d(\mathbf{x}_s - \mathbf{X}_i^k) \Delta S, \quad 1 \leq i \leq P_b^k, \tag{8}$$

$$\mathbf{f} = L^* \mathbf{F}^k, \quad f_s = \sum_{i=1}^{P_b^k} F_i^k d(\mathbf{x}_s - \mathbf{X}_i^k) \Delta l, \quad s \in \Omega_h, \tag{9}$$

где k — индекс погруженной границы Γ_b^k , заданной набором точек \mathbf{X}_i^k , $1 \leq i \leq P_b^k$, F_i^k — значения на погруженной границе в точке X_i^k , \mathbf{f} — значения на множестве Ω_h точек пространственной сетки области Ω площадью (объемом) ячейки ΔS , Δl — характерный элемент длины (площади) границы. Функция $d(\mathbf{r})$ — дискретная аппроксимация дельта-функции на заданной сетке Ω_h , для которой существует широкий выбор способов определения. В работе [11] используется следующая функция, эквивалентная билинейной интерполяции, которую для двумерной области на равномерной сетке с шагом h можно записать следующим образом:

$$d(\mathbf{x}_s - \mathbf{X}_i^k) = \frac{1}{h^2} d_h\left(\frac{x_s - X_i^k}{h}\right) d_h\left(\frac{y_s - Y_i^k}{h}\right),$$

$$d_h(r) = \begin{cases} 1 - |r|, & |r| \leq 1, \\ 0, & |r| > 1. \end{cases} \quad (10)$$

Несмотря на приемлемые результаты, полученные для задач воспроизведения течений со стационарными границами, применение функции (10) в задачах с неоднородными условиями (4) приводит к появлению фиктивных осцилляций в интегральных характеристиках течения [12, 45]. Авторами статьи [12] предлагается преобразование известных $d(\mathbf{r})$, которое позволяет повысить порядок гладкости и обеспечить выполнение сохранения моментов высших порядков этих функций, что приводит к существенному уменьшению амплитуды осцилляций. В настоящей работе для численных расчетов использовалось следующее приближение дельта-функции [46]:

$$d_h(r) = \begin{cases} \frac{1}{3} (1 + \sqrt{-3r^2 + 1}), & |r| < 0.5, \\ \frac{1}{6} (5 - 3|r| - \sqrt{1 - 3(1 - |r|)^2}), & 0.5 \leq |r| \leq 1.5, \\ 0, & |r| > 1.5, \end{cases} \quad (11)$$

а также вариант (11) для задач с подвижными границами, приведенный в работе [12]:

$$d_h(r) = \begin{cases} \frac{17}{48} + \frac{\sqrt{3}\pi}{108} + \frac{|r|}{4} - \frac{r^2}{4} + \frac{1 - 2|r|}{16} \sqrt{-12r^2 + 12|r| + 1} - \frac{\sqrt{3}}{12} \arcsin\left(\frac{\sqrt{3}}{2}(2|r| - 1)\right), & |r| < 1, \\ \frac{55}{48} - \frac{\sqrt{3}\pi}{108} - \frac{13|r|}{12} + \frac{r^2}{4} + \frac{2|r| - 3}{48} \sqrt{-12r^2 + 36|r| - 23} + \frac{\sqrt{3}}{36} \arcsin\left(\frac{\sqrt{3}}{2}(2|r| - 3)\right), & 1 \leq |r| \leq 2, \\ 0, & |r| > 2. \end{cases}$$

Для нахождения функции $\mathbf{f}(\mathbf{x}, t)$ вычисление промежуточного поля скорости \mathbf{u}^* в (5) разделяется на несколько шагов:

$$\frac{\hat{\mathbf{u}} - \mathbf{u}^n}{\Delta t} + [\mathbf{u} \cdot \nabla \mathbf{u}]^{n+1/2} = \nu [\nabla^2 \mathbf{u}]^{n+1/2} - \frac{1}{\rho} \nabla p^n,$$

$$\frac{\mathbf{u}^* - \hat{\mathbf{u}}}{\Delta t} = \mathbf{f}^{n+1}(\mathbf{x}, t_{n+1}) \equiv L^* \mathbf{F}^{n+1}(\mathbf{x}_b, t_{n+1}), \quad (12)$$

где функция $\mathbf{f}^{n+1}(\mathbf{x}, t_{n+1})$ определяется через проекцию с помощью оператора L^* , заданного соотношением (9) соответствующих значений функции $\mathbf{F}^{n+1}(\mathbf{x}_b, t_{n+1})$ на погруженной границе. Применение оператора интерполяции (8) к соотношению (12) приводит к системе уравнений для значений в точках на погруженных границах: $\frac{L \mathbf{u}^* - L \hat{\mathbf{u}}}{\Delta t} = L \mathbf{f}^{n+1} \equiv LL^* \mathbf{F}^{n+1} \equiv A_b \mathbf{F}^{n+1}$, где A_b — симметричная блочно-диагональная квадратная матрица, число блоков равно числу погруженных границ N_b , при этом k -й блок A_b^k матрицы A_b имеет порядок P_b^k . Данные блоки имеют ленточную структуру, которая зависит от распределения точек погруженной границы на вычислительной сетке и используемой аппроксимации дельта-функции. Значения промежуточного поля $L \mathbf{u}^*$ скорости в точках на погруженной границе заменяются известными граничными условиями \mathbf{U}_b^k для скорости на $(n+1)$ -м шаге по времени. Таким образом, вычисление неизвестной добавочной функции $\mathbf{f}^{n+1}(\mathbf{x}, t_{n+1})$ сводится к решению системы линейных уравнений с матрицей A_b относительно $\mathbf{F}^{n+1}(\mathbf{x}_b, t_{n+1})$ с известной правой частью и применению оператора

проектирования (9) к полученному решению:

$$\frac{U_b^k - L \hat{u}}{\Delta t} = A_b F^{n+1}(\mathbf{x}_b, t_{n+1}), \tag{13}$$

$$\frac{\mathbf{u}^* - \hat{u}}{\Delta t} = L^* F^{n+1}(\mathbf{x}_b, t_{n+1}). \tag{14}$$

Существование обратной матрицы A_b следует из отсутствия совпадающих точек в определении погруженных границ. Однако при близком расположении точек Γ_b^k относительно шага сетки возможны проблемы с устойчивостью численного метода решения системы (13). При неравномерном распределении точек погруженной границы и различных значениях Δl необходима модификация операторов (8) и (9).

Предложенный метод нахождения добавочной функции приводит к тому, что краевые условия выполняются точно для промежуточного поля скорости \mathbf{u}^* и лишь приближенно для скорости на $(n + 1)$ -м шаге по времени, полученной проекцией (7) \mathbf{u}^* на основе ограничения для скорости, выраженного уравнением неразрывности. Указанный недостаток характерен для многих модификаций метода погруженной границы [8, 47–49]. Точность аппроксимации можно повысить проведением нескольких итераций на каждом шаге по времени. В работах [50, 51] описываются неявные методы, в которых требуется одновременное выполнение уравнения неразрывности и краевых условий для скорости \mathbf{u}^{n+1} , однако применение предложенных схем для задач с подвижными границами требует значительных вычислительных ресурсов.

Следует отметить, что приведенный способ вычисления добавочной функции в методе погруженной границы не накладывает ограничений на возможность применения неявных схем для дискретизации по времени диффузионных слагаемых в уравнении движения (например, схемой Кранка–Николсон второго порядка), что важно при моделировании вязких течений с малым числом Рейнольдса. Для повышения точности решения в задачах обтекания тел широко применяются сгущенные сетки вблизи погруженных границ. Обобщение данного подхода в этом случае достаточно просто, если возможно выделение некоторой подобласти неравномерной сетки, включающей в себя погруженную область, на которой шаг сетки остается постоянным, иначе необходимо переопределение приближений дельта-функций на сетках с неравномерным шагом (один из способов предложен в работе [52]). В сравнении с другими вариантами метода погруженной границы приведенная методика лучше подходит для решения задач с нестационарной геометрией области и требует лишь пересчета операторов интерполяции (8) и (9) и матрицы A_b на каждом шаге по времени. Важной особенностью является легкость вычисления следующих интегральных характеристик течения по известной функции $\mathbf{f}(\mathbf{x}, t)$: коэффициента лобового сопротивления C_D и коэффициента подъемной силы C_L . В двумерном случае для стационарных погруженных границ коэффициенты C_D и C_L определяются следующим образом:

$$C_D = \frac{2F_D}{U_c^2 L_c}, \quad F_D = - \int_{\Omega} f_x(\mathbf{x}) d\mathbf{x} = - \sum_{\mathbf{x} \in \Omega_h} f_x(\mathbf{x}) \Delta s, \tag{15}$$

$$C_L = \frac{2F_L}{U_c^2 L_c}, \quad F_L = - \int_{\Omega} f_y(\mathbf{x}) d\mathbf{x} = - \sum_{\mathbf{x} \in \Omega_h} f_y(\mathbf{x}) \Delta s, \tag{16}$$

где U_c, L_c — характеристическая скорость течения и характеристическая длина криволинейной границы соответственно, $f_x(\mathbf{x}), f_y(\mathbf{x})$ — компоненты вектора $\mathbf{f}(\mathbf{x}, t)$ по осям x и y , определяемые как решение системы (13).

3. Программная реализация на графических процессорах. Применение явной схемы для дискретизации системы уравнений Навье–Стокса и схем второго порядка аппроксимации пространственных производных приводит к тому, что основная доля расчетов приходится на решение уравнения Пуассона (6) и на арифметические операции, связанные с методом погруженной границы (13)–(16).

Для решения конечно-разностного уравнения Пуассона (6) во всей области Ω_h с симметричной матрицей в данной работе использовался предобусловленный метод сопряженных градиентов. Выбор метода сопряженных градиентов для численного решения эллиптического уравнения позволяет также оценить эффективность параллельной реализации на графических процессорах и других итерационных методов градиентного типа, которые широко применяются во многих численных моделях вычислительной гидродинамики.

Систему линейных уравнений, соответствующую дискретизации уравнения (6) центральными разностями второго порядка, можно записать в виде

$$A_p x = b_p, \quad A_p = L + D + L^T, \tag{17}$$

где D — диагональная, L — нижняя треугольная часть матрицы A_p без учета диагональных элементов. Выбор подходящего предобусловливателя M для матрицы системы ($A_p = M + N$, $Nx \approx 0$) в методе сопряженных градиентов приводит к уменьшению числа обусловленности и, как следствие, к существенному увеличению скорости сходимости. В большинстве работ, посвященных реализации на графических процессорах итерационных методов, рассматривается предобусловливатель Якоби или его блочный аналог [25]. Такой подход позволяет эффективно использовать особенности архитектуры, однако сильно проигрывает более сложным предобусловливателям относительно уменьшения числа обусловленности системы уравнений. Для предобусловливателей вида $M = LU$ (L — нижняя, U — верхняя треугольные матрицы) в общем случае разреженной матрицы A возникают трудности с реализацией быстрого обращения матрицы M на графических процессорах и распределением вычислений на отдельные параллельные потоки. Подходящим выбором для вычислений на графических устройствах представляются методы [21, 22], в которых считается, что известна аппроксимация обратной матрицы системы: $M^{-1} \approx A_p^{-1}$. В этом случае дополнительные операции, вызванные переходом к предобусловленной системе, сводятся к умножению матрицы на вектор. Тем не менее, как правило, обратная матрица приближается разреженной матрицей, что по мере усложнения структуры исходной матрицы приводит к заметному ухудшению результатов [14].

При решении системы (17) с матрицей A_p в условиях задачи (1)–(4) сравнивались реализации на графических процессорах нескольких вариантов предобусловливателей:

$$M_J = D, \quad (18)$$

$$M_{SGS} = (D + L) D^{-1} (D + L)^T, \quad (19)$$

$$M_{AIR}^{-1} = (I - LD^{-1}) (I - D^{-1}L^T), \quad (20)$$

где M_J — соответствует методу Якоби, M_{SGS} — симметричному методу Гаусса–Зейделя, а предобусловливатель M_{AIR} предложен в работе [26] для решения сеточного уравнения Пуассона и относится к методам аппроксимации обратной матрицы. Для реализации предобусловливателя M_{SGS} (19) на графических процессорах применялось переупорядочивание узлов сетки в шахматном порядке. Такой предобусловливатель соответствует симметричному методу последовательной верхней релаксации с оптимальным значением параметра релаксации, равным единице [20]. В качестве предобусловливателя также рассматривался геометрический многосеточный метод с V-циклом, где симметричный метод Гаусса–Зейделя с переупорядочиванием узлов сетки использовался как вспомогательный итерационный метод. Для проверки сходимости метода сопряженных градиентов использовалось условие малости относительной погрешности нормы невязки.

Важной особенностью архитектуры графических процессоров является существенная неоднородность типов памяти устройств. Одна из проблем при реализации предобусловленного метода сопряженных градиентов на данной архитектуре — большое число точек синхронизации вычислений, что приводит к увеличению операций чтения/записи для данных в медленной глобальной памяти устройства и сокращению числа вычислений в быстрой памяти. Повысить быстродействие итерационного метода на графических процессорах возможно за счет группировки вычислений для уменьшения обращений к медленной памяти и выполнения асинхронных операций копирования памяти устройства. Модификации метода сопряженных градиентов [18, 19] позволяют использовать часть этих особенностей архитектуры. Заметного улучшения производительности удалось добиться при сочетании предобусловливателя (20) и варианта метода сопряженных градиентов, предложенного авторами статьи [19]. В алгоритме [19] предполагается, что предобусловливатель можно записать в виде $M = LL^T$, при этом обращение матрицы M разбивается на несколько шагов, а обновление значения вектора приближенного решения откладывается на следующую итерацию. В случае если матрица M определяется соотношением (20), то решение системы с матрицей предобусловливателя заменяется умножением нижнетреугольной и верхнетреугольной матриц на вектор. Такой подход позволяет совместить вычисления и значительную часть обменов данными между центральным и графическим процессором и взаимодействие между отдельными графическими устройствами.

Для решения системы линейных уравнений с симметричной матрицей A_b (13) при аппроксимации краевых условий методом погруженной границы использовалась модификация метода сопряженных градиентов [18] с предобусловливателем Якоби, что позволяет проводить единственный обмен данными с устройством на каждой итерации. При этом вариант [18] менее устойчив по сравнению с классическим методом и обладает большей зависимостью от распределения точек погруженной границы на сетки, характеризуемой соотношением $\Delta l/h$, где h — шаг сетки и Δl — элемент длины (площади) криволинейной границы. В приводимых расчетах полагалось $\Delta l/h \approx 0.785$, что не приводит к существенному увеличению числа итераций, достаточных для сходимости метода [18]. Однако при уменьшении значения $\Delta l/h$ в ряде

численных расчетов наблюдалось заметное увеличение числа итераций, нехарактерное для классического метода сопряженных градиентов. Число узлов сетки намного превосходит число точек, выбранных для описания криволинейных границ, что приводит к тому, что аппроксимацию краевых условий (13) и (14) и определение коэффициентов (15) и (16) для некоторых задач эффективнее проводить на центральном процессоре. Тем не менее, разделение части вычислений требует дополнительных операций копирования из памяти устройства и существенно сказывается на производительности. Наиболее предпочтительным представляется выполнение всех численных расчетов на графическом процессоре. Данные, связанные с погруженными границами, группировались в один массив, и для всех криволинейных границ решалась система (13) с блочно-диагональной матрицей, что необходимо для достаточной загруженности вычислительных элементов. В некоторых задачах, возможно, более эффективным было бы разделение погруженных областей по числу точек и значению $\Delta l/h$.

Для организации вычислений на графических процессорах была применена технология программирования CUDA. Расчеты проводились на кластере из восьми графических процессоров Nvidia Tesla T10 второго поколения [http://tesla.parallel.ru] и на суперкомпьютере “ГрафИТ!/GraphIT!” [http://gpu.parallel.ru/graphit.html], где для расчетов использовалось до 15 карт Nvidia “Fermi” Tesla M2050. Обмены данными при работе с несколькими устройствами проводились с помощью функций библиотеки MPI. Рассматривалась декомпозиция двумерной области на вычислительные узлы по одной и двум координатам. На каждой графической карте в рамках среды CUDA потоки группируются в сетку, состоящую из одномерных или двумерных блоков в зависимости от типа операции. Число нитей, объем общей памяти для нитей блока и число регистров подбирались на основе численных экспериментов для отдельных функций. Общую схему вычислений, выполняемых на графическом процессоре, можно описать следующим образом: копирование значений из глобальной памяти в быструю память (регистры, общая память для нитей блока), проведение вычислений, запись результатов в глобальную память. Для проведения сравнений и оценки быстродействия расчетов на графических процессорах используется реализация предложенного численного метода на центральном процессоре (Intel Xeon 5472 3.0Hz, установленном на СКИФ МГУ “Чебышёв”). Распределение вычислений на ядра процессора реализовано с помощью библиотеки OpenMP.

4. Численные эксперименты. Численно моделировался ряд двумерных течений вязкой несжимаемой жидкости: течение вокруг неподвижного кругового цилиндра без учета пограничного слоя (п. 4.1), течение вокруг кругового цилиндра, расположенного несимметрично в канале с твердыми стенками (п. 4.2), течение вокруг группы круговых цилиндров (п. 4.3), а также течение вокруг кругового цилиндра, совершающего вынужденные колебания (п. 4.4). Область Ω из уравнений (1)–(4) в расчетах представляет собой прямоугольник, юго-западный угол которого совпадает с началом координат. На входе в область (западная сторона Γ_W области Ω) задается профиль вектора скорости $\mathbf{u} = (u, v)$:

$$u = U_I(y), \quad v = 0. \tag{21}$$

На выходе из области (восточная сторона Γ_E области Ω) используется следующее граничное условие:

$$\frac{\partial \mathbf{u}}{\partial t} + U_0 \frac{\partial \mathbf{u}}{\partial x} = 0, \tag{22}$$

где U_0 определяет скорость переноса, которая вычисляется осреднением по контуру Γ_E значений продольной компоненты скорости [39]. Условия (22) для численного моделирования выбранных задач более предпочтительны, что обусловлено нестационарным характером рассматриваемых течений, при этом граница Γ_E в расчетах предполагалась достаточно удаленной от криволинейных границ для уменьшения влияния краевых условий на характеристики воспроизводимых потоков.

Число Рейнольдса можно определить следующим образом:

$$Re = \frac{U_c L_c}{\nu}, \tag{23}$$

где L_c — характеристическая длина, которая в численных экспериментах принималась равной диаметру кругового цилиндра D , U_c — характеристическая скорость, которая определяется как среднее значение функции $U_I(y)$ в условии (21). Точное определение краевых условий приводится далее при описании численных экспериментов.

4.1. Течение вокруг кругового цилиндра. Течение вокруг кругового цилиндра при различных числах Рейнольдса исследовалось во многих работах как с помощью численного моделирования, так и проведением физических экспериментов, что делает его надежным способом проверки метода погруженной границы для воспроизведения течения в достаточно сложной области. Данный численный эксперимент также использовался для демонстрации быстродействия реализации на графических процессорах и сравнений с традиционной архитектурой.

На входе в область задается постоянное значение скорости (21): $U_1(y) = U_\infty \equiv \text{const}$. На южной и северной стороне прямоугольной области Ω задаются следующие краевые условия: $\frac{\partial u}{\partial y} = 0, v = 0$.

Вычисления проводились при числе Рейнольдса $Re = 100$, где в определении числа Рейнольдса (23) принимается $U_c \equiv U_\infty$. При малых числах Рейнольдса ($Re < Re_c, Re_c \approx 47$) рассматриваемое течение является стационарным, при этом позади кругового цилиндра образуются два симметричных вихря, продольный размер которых зависит от числа Рейнольдса. При увеличении $Re > Re_c$ течение становится нестационарным и периодическим, позади цилиндра формируется дорожка Кармана.

Для сравнения метода погруженной границы с другими подходами рассчитывались значения коэффициентов лобового сопротивления (15), коэффициента подъемной силы (16) и число Струхала St , соответствующее безразмерной величине частоты периодического течения:

$$St = \frac{f_q}{U_\infty D}, \tag{24}$$

где f_q — частота переменного вихря. В статье [63] предлагается следующая эмпирическая зависимость частоты Струхала от числа Рейнольдса при $Re < Re_0, Re_0 \approx 180$, основанная на экспериментальных данных:

$$St = 0.00016Re + 0.1816 - 3.3265/Re. \tag{25}$$

Вычисления проводились на равномерной прямоугольной сетке в области $[40D, 20D]$, центр кругового цилиндра расположен в точке $(10D, 10D)$. В табл. 1 приведены полученные значения C_D, C_L и числа Струхала при счете на графическом процессоре с одинарной точностью и шагом сетки $0.025D$ в сравнении с другими работами, а также значение числа Струхала на основе зависимости (25). Для определения частоты f_q в (24) использовалось преобразование Фурье для ряда значений C_L по времени.

Основная доля вычислений при численном воспроизведении течения вокруг кругового цилиндра приходится на решение уравнения Пуассона. Так, при решении системы (17) с использованием многосеточного метода в качестве предобусловливателя доля вычислений, связанных с операциями метода погруженной границы, составляет несколько процентов.

На рис. 1 показано ускорение вычислений на ГПУ (рис. 1а — Nvidia Tesla T10, рис. 1б — Nvidia Tesla T20) относительно четырех ядер центрального процессора Intel Xeon E5472 3.0Hz в зависимости от типа предобусловливателя в методе сопряженных градиентов. Приняты следующие обозначения: линия 1 — предобусловливатель Якоби (18); линия 2 — симметричный метод Гаусса–Зейделя (19) с переупорядочиванием узлов; линия 3 — предобусловливатель (20); линия 4 — геометрический многосеточный метод с V-циклом.

Таблица 1
Значения C_D (среднее), C_L (максимум–минимум), * — среднее квадратическое), St

	C_D	C_L	St
Williamson [63], 1989	—	—	0.164
Braza [57], 1986	1.36 ± 0.015	± 0.25	—
Liu [58], 1998	1.35 ± 0.012	± 0.339	0.164
Tseng [3], 2003	1.42	0.29*	0.164
Lai [6], 2000	1.4473	0.3299*	0.165
Calhoun [59], 2002	1.330 ± 0.014	± 0.298	0.175
Herfjord [60], 1996	1.36	± 0.34	0.168
Berthelsen [61], 2008	1.38 ± 0.01	± 0.34	0.169
Результаты вычислений	1.405 ± 0.0072	± 0.348	0.168

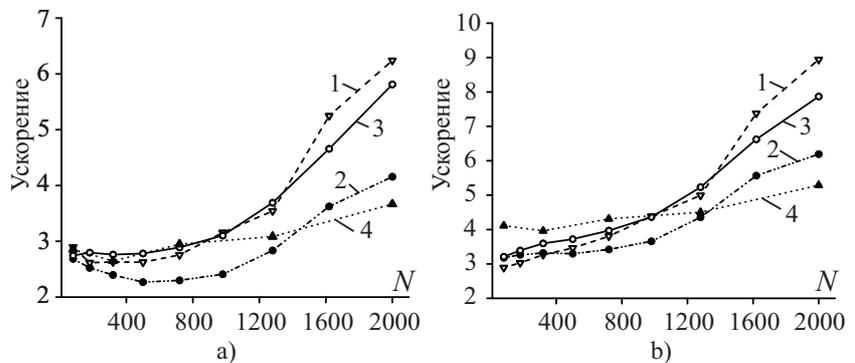


Рис. 1. Ускорение вычислений на графическом процессоре в зависимости от типа предобусловливателя:
а) Nvidia Tesla T10, б) Nvidia Tesla T20

Вычисления проводились с одинарной точностью на последовательности сеток с уменьшением шага h . Значение N по горизонтальной оси соответствует числу точек сетки (в тысячах). Отношение шага сетки h к шагу по времени и отношение элемента длины кругового цилиндра Δl к h в расчетах полагались постоянными. Ускорение рассчитывалось как отношение осредненного времени счета на центральном и графическом процессоре для одного шага интегрирования полной системы за промежуток времени, составляющий один период вихревой дорожки Кармана.

Несмотря на значительное ускорение для предобусловливателя Якоби (18) на графическом процессоре, применение других подходов позволяет сократить общее время счета. Для симметричного метода Гаусса–Зейделя время счета уменьшается на 35% для сетки с шагом $h = 0.025D$. Низкое ускорение симметричного метода Гаусса–Зейделя с переупорядочиванием узлов на графических процессорах связано с увеличением числа точек синхронизации и необходимостью в выполнении медленных операций чтения/записи глобальной памяти для двух наборов из чередующихся точек сетки в операциях матрично-векторного умножения. Существенного снижения производительности можно ожидать в том случае, если для параллельной реализации симметричного метода Гаусса–Зейделя требуется многоцветное упорядочивание узлов сетки. Одним из способов решения данной проблемы является изменение способа хранения значений векторных переменных с учетом способа переупорядочивания узлов сетки [27], однако точная оценка преимуществ реализации такого подхода на графических процессорах требует дополнительных исследований.

Для предобусловливателя (20) и модификации метода сопряженных градиентов [19] наблюдается аналогичное сокращение времени счета относительно предобусловливателя Якоби. Тем не менее, для трехмерных задач и при использовании схем более высокого порядка по пространству можно ожидать, что матрица предобусловливателя (20) будет хуже аппроксимировать обратную матрицу системы уравнений, что может привести к падению точности результатов по сравнению с другими предобусловливателями [14].

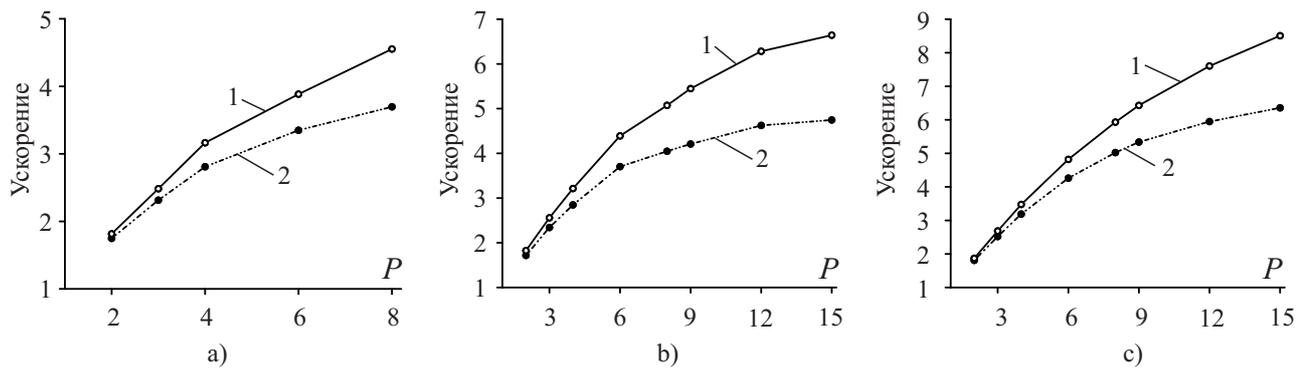


Рис. 2. Ускорение при распределении вычислений по графическим процессорам: а) Nvidia Tesla T10, $h = 0.025D$; б) Nvidia Tesla T20, $h = 0.01667D$; в) Nvidia Tesla T20, $h = 0.0125D$

В приводимых расчетах для многосеточного метода (рис. 1, линия 4) число сеток последовательно увеличивается так, что число точек на самой грубой сетке остается приблизительно постоянным при масштабировании задачи, при этом решение системы на наиболее грубых сетках в силу малого числа точек эффективнее проводить на центральном процессоре. Быстродействие реализации многосеточного метода на графическом процессоре существенно зависит от объема вычислений и, как следствие, уменьшается при увеличении шага сетки. Данный эффект будет более существенным при реализации многосеточного метода с W- или F-циклом [15]. Следует заметить, что в настоящей работе использовались достаточно простые операторы проектирования и продолжения. Применение более сложных и точных вариантов этих операторов должно приводить к увеличению ускорения реализации на графических процессорах относительно традиционной архитектуры.

На рис. 2 представлены результаты распределения вычислений по графическим процессорам с помощью библиотеки MPI для предобусловливателя (20) (рис. 2, линия 1) и симметричного метода Гаусса–Зейделя (19) с переупорядочиванием узлов сетки (рис. 2, линия 2). Рассматривалась масштабируемость задачи о течении вокруг кругового цилиндра на кластере из 8 процессоров Nvidia Tesla T10 для сетки с шагом $h = 0.025D$ (рис. 2a) и на суперкомпьютере “ГрафИТ!” (Nvidia Tesla T20) для сеток с шагом $h = 0.01667D$ (рис. 2b) и $h = 0.0125D$ (рис. 2c). При реализации предобусловливателя (20) умножение на матрицу M^{-1} в модификации [19] выполнялось без синхронизации, что в отличие от симметричного метода Гаусса–Зейделя не приводит к существенному увеличению числа итераций в методе сопряженных градиентов. Переупорядочивание узлов сетки для предобусловливателя (19) вызывает дополнительные

трудности при организации обменов данными между графическими процессорами.

Сравнение результатов при декомпозиции расчетной области по одной и двум координатам на вычислительные узлы показало, что в последнем случае наблюдается падение производительности. К основным причинам можно отнести увеличение числа обменов, приходящихся на узел, что приводит к увеличению числа медленных операций копирования графической памяти, и отсутствие последовательного доступа к памяти устройства по одной координате.

4.2. Течение вокруг кругового цилиндра в канале. Рассматривается задача о течении вокруг кругового цилиндра в канале с твердыми стенками в постановке, предложенной в работе [64]. Расчетная область $\Omega = [22D, 4.1D]$, центр кругового цилиндра смещен относительно центра канала и расположен в точке $(2D, 2D)$. На входе области задается параболический профиль скорости (21): $U_1(y) \equiv 6y(H-y)/H^2$, где $H = 4.1D$ — высота канала. На стенках канала (южная и северная сторона области Ω) требуется выполнения условия прилипания: $u = 0, v = 0$. Число Рейнольдса определяется через среднее значение скорости $U_c \equiv 2U_1(H/2)/3$ и составляет $Re = 100$, что соответствует нестационарному течению.

В работе [64] представлены подробные результаты численного моделирования на основе различных методов дискретизации уравнений Навье–Стокса с измельчением шага сетки и шага интегрирования уравнений по времени. По полученным данным выводится интервал допустимых значений для коэффициента лобового сопротивления C_D , коэффициента подъемной силы C_L и числа Струхала St . Данный эксперимент позволяет получить более точную оценку способности метода погруженной границы воспроизводить течения в областях сложной конфигурации, а также погрешности вычислений интегральных характеристик с одинарной и двойной точностью на графическом процессоре. В табл. 2 приведены рассчитанные значения коэффициентов C_D и C_L на графическом и центральном процессорах с одинарной и двойной точностью. Вычисления проводились на сетках с шагом сетки $h = 0.05D, 0.025D, 0.0125D$. Число Куранта и отношение $\Delta t/h$ оставались постоянными при измельчении сетки. Число Струхала для всех расчетов составляло $St = 0.3 \pm 5 \times 10^{-5}$, что хорошо согласуется с интервалом, приведенным в работе [64]: $0.2950 \leq St \leq 0.3050$.

Таблица 2

Значения C_D (максимум), C_L (максимум)

	h	C_D	C_L
Nvidia Tesla M2050, одинарная точность	$0.05D$	3.2178	0.9123
Nvidia Tesla M2050, одинарная точность	$0.025D$	3.2461	0.9671
Nvidia Tesla M2050, двойная точность	$0.0125D$	3.2544	0.9946
Intel Xeon E5472 3.0Hz, одинарная точность	$0.025D$	3.2462	0.9670
Intel Xeon E5472 3.0Hz, двойная точность	$0.025D$	3.2461	0.9668
Интервал допустимых значений [64]		3.22–3.24	0.99–1.01

При вычислениях с двойной точностью на графическом процессоре (Nvidia Tesla M2050) время счета увеличивается в 1.5–1.8 раз относительно расчетов с одинарной точностью в зависимости от размерности сетки, при этом разность в значениях интегральных характеристик совпадает по порядку с разницей значений, приводимых в табл. 2 для центрального процессора.

4.3. Течение вокруг группы круговых цилиндров. Увеличение числа погруженных областей и, как следствие, увеличение числа точек криволинейной границы на заданной сетке может привести к тому, что время решения системы (13), применения операторов (8), (9) и вычисления коэффициентов (15), (16) становится соизмеримым со временем, затраченным на решение уравнения Пуассона (6), что связано с неравномерным ускорением алгоритма на архитектуре графических процессоров. Декомпозиция области по графическим процессорам с помощью библиотеки MPI также требует выполнения дополнительных обменов данными между процессами при аппроксимации краевых условий на криволинейных границах.

Для оценки быстродействия реализации на графических процессорах метода погруженной границы численно решалась задача о течении вокруг группы из N_b ($N_b = 3, \dots, 18$) круговых цилиндров. В большинстве работ аналогичные задачи решаются для анализа эффективности предлагаемых методов и алгоритмов. Например, в статье [56] течение вокруг группы круговых цилиндров исследуется для проверки многосеточного метода в области с погруженными границами. В работе [55] проводилось моделирование течения вокруг группы из восемнадцати круговых цилиндров, расположенных в шахматном порядке, при числах Рейнольдса, равных 100 и 200. Из численных расчетов [55] следует, что течение сохраняет

стационарность при $Re = 100$ и становится нестационарным при $Re = 200$.

Конфигурация области для задачи выбиралась сходной работе [55]: $\Omega = [45D, 30D]$. Краевые условия для области Ω совпадают с краевыми условиями для течения вокруг кругового цилиндра (п. 4.1). Центры круговых цилиндров в каждом столбце имеют следующие координаты по x : $x_i = 10D + 3(i - 1)D/2$, $1 \leq i \leq 7$. Для нечетных значений i координата y центра круговых цилиндров: $y_{ij} = 12D + 3(j - 1)D$, $1 \leq j \leq 3$, для четных i : $y_{ij} = 13.5D + 3(j - 1)D$, $1 \leq j \leq 2$. Вычисления проводились на равномерной прямоугольной сетке с шагом $h = 0.05D$ и $h = 0.025D$.

На рис. 3 приведено ускорение, полученное на одном графическом процессоре (Nvidia Tesla M2050) относительно вычислений на одном ядре Intel Xeon E5472 3.0Hz для расчетов, связанных с методом погруженной границы (13) и (14), и вычисления коэффициентов (15) и (16) при последовательном добавлении столбцов круговых цилиндров на фиксированной сетке с шагами $h = 0.05D$ (рис. 3а) и $h = 0.025D$ (рис. 3б). Значения по горизонтальной оси N_b определяет число круговых цилиндров. Результаты соответствуют решению системы (13) классическим методом сопряженных градиентов (рис. 3, линия 2) и его модификацией [18] (рис. 3, линия 1). При $N_b = 18$ доля операций (13)–(16) от общего времени интегрирования системы на один шаг по времени при использовании многосеточного метода в качестве предобусловливателя в итерационном методе решения уравнения Пуассона составляет приблизительно 20%. Вычисление коэффициента лобового сопротивления и коэффициента подъемной силы на графическом процессоре проводились отдельно для каждой погруженной границы, что при $N_b > 8$ занимает 10–20% от всех расчетов (13)–(16). Учитывая относительно малое число точек погруженных границ, улучшение результатов при использовании варианта метода сопряженных градиентов [18] достигается за счет экономии времени, необходимого для инициализации операции копирования графической памяти.

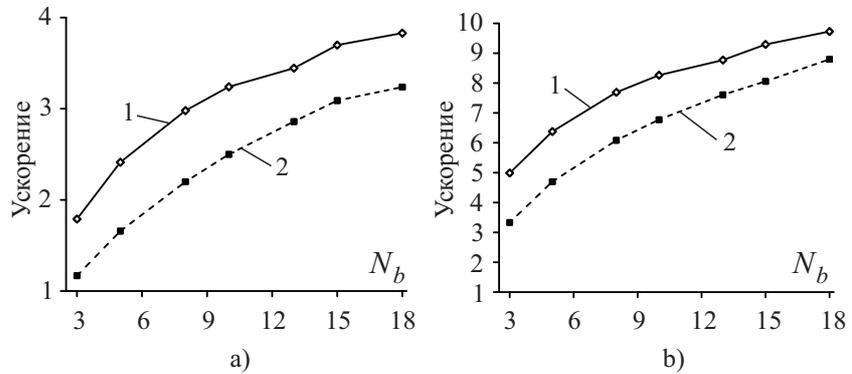


Рис. 3. Ускорение операций метода погруженной границы на графическом процессоре: а) $h = 0.05D$; б) $h = 0.025D$

Для случая $N_b = 18$ также проводилась серия вычислений при числах Рейнольдса в интервале от 100 до 200. Полученные результаты свидетельствуют о том, что течение сохраняет стационарность при числах Рейнольдса $100 \leq Re < 140$ на достаточно долгих временных интервалах ($T \gg T_s$, где T_s — время установления периодического течения при моделировании течения вокруг единичного кругового цилиндра (п. 4.1) при $Re = 100$). При дальнейшем увеличении времени интегрирования наблюдается рост коэффициента лобового сопротивления и малые устойчивые колебания значений C_D и C_L , что характерно для нестационарного режима. Данное обстоятельство может быть связано с тем, что вычисления проводились на графическом процессоре с одинарной точностью. Более точное определение характеристик течения при $N_b = 18$ и нахождение критического числа Рейнольдса требует дополнительных исследований.

4.4. Течение вокруг подвижного кругового цилиндра. Рассматривается течение вокруг кругового цилиндра, совершающего гармонические колебания в поперечном направлении к потоку. Движение цилиндра задается законом

$$y(t) = A \sin(2\pi f_0 t). \tag{26}$$

В расчетах принимались следующие значения: амплитуда $A = 0.2D$, частота $f_0 = 0.8f_s$, $f_0 = 1.1f_s$, где f_s — частота дорожки Кармана для стационарной задачи при $Re = 185$. К параметрам, определяющим характеристики данного течения, можно отнести отношение частот f_0/f_s и амплитуды колебаний к диаметру цилиндра. При приближении частоты f_0 к частоте f_s наблюдается синхронизация движения цилиндра с вихревой дорожкой, образующейся позади цилиндра. Экспериментальные исследования [66] показали, что в этом случае происходит увеличение средних значений коэффициентов лобового сопротивления и коэффициента подъемной силы. Для сил, действующих на цилиндр, наблюдается отклонение в фазе относительно колебаний, заданных соотношением (26). В работе [65] получен следующий диапазон частот, при котором наблюдается эффект синхронизации: $0.85 < \frac{f_0}{f_s} < 1.17$.

Конфигурация области, положение цилиндра и краевые условия совпадают с задачей воспроизведе-

ния течения вокруг стационарного кругового цилиндра (п. 4.1).

В табл. 3 приведены значения для коэффициента лобового сопротивления и коэффициента подъемной силы, полученные при вычислениях на графическом процессоре с одинарной точностью на сетке с шагом $h = 0.025D$. На рис. 4 приведены графики C_D , C_L по времени для соотношений частот $f_0 = 0.8f_s$ (рис. 4a) и $f_0 = 1.1f_s$ (рис. 4b).

Таблица 3
Значения C_D (среднее), C_L (среднее квадратическое)

	C_D	C_L	C_D	C_L
	$f_0 = 0.8f_s$	$f_0 = 0.8f_s$	$f_0 = 1.1f_s$	$f_0 = 1.1f_s$
Guilmineau [67], 2002	1.195	0.08	1.42	0.897
Yang [12], 2009	1.281	0.076	—	—
Wu [62], 2008	—	—	1.454	0.854
Результаты вычислений	1.267 ± 0.039	0.063	1.457 ± 0.155	0.943

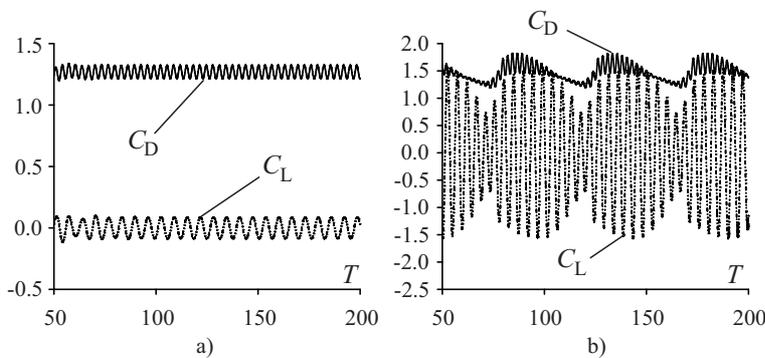


Рис. 4. Временной ряд значений C_D и C_L :
а) $f_0 = 0.8f_s$; б) $f_0 = 1.1f_s$

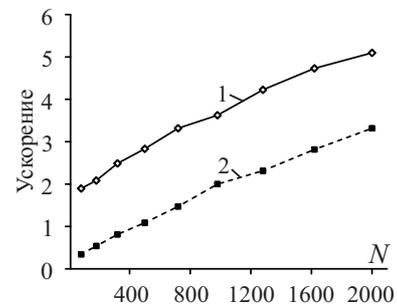


Рис. 5. Ускорение операций метода погруженной границы на графическом процессоре

Ускорение реализации на графическом процессоре (Nvidia Tesla M2050) относительно одного ядра центрального процессора Intel Xeon E5472 3.0Hz для операций метода погруженной границы (13)–(16) и при пересчете матрицы системы (13) при вынужденных осцилляциях кругового цилиндра показано на рис. 5 (рис. 5, линия 1). Для сравнения приводится ускорение вычислений (13)–(16) для неподвижного цилиндра (рис. 5, линия 2). График построен при увеличении числа точек криволинейной границы и одновременном масштабировании прямоугольной сетки. Число Куранта и отношение Δl к шагу сетки h оставалось постоянным. При моделировании подвижных границ увеличивается доля вычислений, связанных с аппроксимацией краевых условий, что может значительно влиять на общее время численного эксперимента. Рост числа точек сетки (с сохранением соотношения $\Delta l/h$) приводит к ожидаемому уменьшению (в силу различия в размерности числа точек границы и размерности числа точек двухмерной сетки) доли вычислений, приходящихся на операции (13)–(16), и на пересчет матрицы (13) при смещении границы. Тем не менее, поскольку операторы (8), (9) также зависят от разрешения сетки, это уменьшение не так значительно. Для рассматриваемой задачи расчеты, связанные с методом погруженной границы, на графическом процессоре составили от 20 до 35 процентов от общего числа вычислений в зависимости от размерности сетки (при решении уравнения Пуассона в качестве предобусловливателя использовался геометрический многосеточный метод).

5. Заключение. В настоящей статье рассмотрена реализация метода погруженной границы для графических процессоров. На примере решения ряда гидродинамических задач показаны основные особенности реализации численного метода для решения уравнений Навье–Стокса на архитектуре графических процессоров. Результаты расчетов свидетельствуют о возможности эффективного моделирования течений в областях сложной конфигурации на данной архитектуре и применения метода погруженной границы для описания криволинейных границ на прямоугольных сетках. При этом удается добиться ускорения всех основных частей численного метода, в том числе ускорения итерационного метода решения эллипти-

ческого уравнения и метода погруженной границы. Существенной проблемой при распределении данных на группу графических процессоров являются дополнительные операции копирования памяти устройства, что необходимо учитывать при решении трехмерных задач, где увеличивается объем пересылаемых данных и число графических карт, необходимых для моделирования течений с достаточным пространственным разрешением.

Важной задачей представляется разработка и исследование “быстрых” предобусловливателей для итерационных методов на графических процессорах, в частности для трехмерных эллиптических уравнений. Для некоторых вариантов метода погруженной границы в ряде случаев многосеточные методы менее эффективны, а использование простейших предобусловливателей в итерационном методе приводит лишь к незначительным улучшениям. Вопросы, связанные с быстродействием геометрического многосеточного метода для распределенных вычислений с дополнительным параллелизмом за счет графических процессоров, также требуют специальных исследований. Интересным представляется разработка реализаций на графических процессорах моделей численного воспроизведения турбулентных течений, таких как методы вихреразрешающего моделирования [68, 69], в которых, как правило, значительная доля вычислений приходится на операции, связанные с замыканием системы уравнений.

Работа выполнена при поддержке победителей первого и второго этапов конкурса компании “Т-Платформы” совместно с МГУ им. М. В. Ломоносова “Эффективное использование GPU-ускорителей при решении больших задач”.

СПИСОК ЛИТЕРАТУРЫ

1. *Mittal R., Iaccarino G.* Immersed boundary methods // *Annual Review of Fluid Mechanics.* 2005. **37**. 239–261.
2. *Peskin C.S.* The fluid dynamics of heart valves: experimental, theoretical and computational methods // *Annual Review of Fluid Mechanics.* 1982. **14**. 235–259.
3. *Tseng Y.-H., Ferziger J.H.* A ghost-cell immersed boundary method for flow in complex geometry // *J. of Computational Physics.* 2003. **192**, N 2. 593–623.
4. *Mohd-Yusof J.* Combined immersed boundary/B-spline methods for simulation of flow in complex geometries // *CTR Annual Research Briefs, Center for Turbulence Research.* Stanford: Stanford University Press, 1997. 317–328.
5. *Goldstein D., Handler R., Sirovich L.* Modeling a no-slip flow boundary with an external force field // *J. of Computational Physics.* 1993. **105**, N 2. 354–366.
6. *Lai M.-C., Peskin C.S.* An immersed boundary method with formal second-order accuracy and reduced numerical viscosity // *J. of Computational Physics.* 2000. **160**, N 2. 705–719.
7. *Saiki E.M., Biringen S.* Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method // *J. of Computational Physics.* 1996. **123**. 450–465.
8. *Fadlun E.A., Verzicco R., Orlandi P., Mohd-Yusof J.* Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations // *J. of Computational Physics.* 2000. **161**, N 1. 35–60.
9. *Balaras E.* Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations // *Computers and Fluids.* 2004. **33**, N 3. 375–404.
10. *Yang J.M., Balaras E.* An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries // *J. of Computational Physics.* 2006. **215**, N 1. 12–40.
11. *Su S.-W., Lai M.-C., Lin C.-A.* An immersed boundary technique for simulating complex flows with rigid boundary // *Computers and Fluids.* 2007. **36**, N 2. 313–324.
12. *Yang X., Zhang X., Li Z., He G.-W.* A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations // *J. of Computational Physics.* 2009. **228**, N 20. 7821–7836.
13. *Van der Vorst H.A.* Iterative Krylov methods for large linear systems. Cambridge: Cambridge University Press, 2003.
14. *Dongarra J.J., Duff I.S., Sorensen D.C., Van der Vorst H.A.* Numerical linear algebra for high-performance computers. Philadelphia: SIAM, 1998.
15. *Ольшанский М.А.* Лекции и упражнения по многосеточным методам. М.: Изд-во Моск. ун-та, 2003.
16. *Шайдулов В.В.* Многосеточные методы конечных элементов. М.: Наука, 1989.
17. *Wesseling P.* An introduction to multigrid methods. New York: Wiley, 1992.
18. *Chronopoulos A.T., Gear C.W.* S-step iterative methods for symmetric linear systems // *J. of Computational and Applied Mathematics.* 1989. **25**, N 2. 153–168.
19. *Demmel J., Heath J., Van der Vorst H.* Parallel numerical linear algebra // *Acta Numerica.* 1993. **2**. 111–197.
20. *Harrar D.L., Ortega J.M.* Optimum m-step SSOR preconditioning // *J. of Computational and Applied Mathematics.* 1988. **24**, N 1-2. 195–198.
21. *Benzi M., Meyer C.D., Tuma M.* A sparse approximate inverse preconditioner for the conjugate gradient method // *SIAM J. on Scientific Computing.* 1996. **17**, N 5. 1135–1149.
22. *Cosgrove J.D.F., Dias J.C., Griewank A.* Approximate inverse preconditionings for sparse linear systems // *Int. J. of Computer Mathematics.* 1992. **44**, N 1-4. 91–110.

23. Molemaker J., Cohen J.M., Patel S., Noh J. Low viscosity flow simulations for animation // Proc. of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Aire-la-Ville: Eurographics Associations, 2008. 9–18.
24. Bolz J., Farmer I., Grinspun E., Schröder P. Sparse matrix solvers on the GPU: conjugate gradients and multigrid // ACM Transactions on Graphics. 2003. **22**, N 3. 917–924.
25. Buatois L., Caumon G., Levy B. Concurrent number cruncher: a GPU implementation of a general sparse linear solver // Int. J. of Parallel, Emergent and Distributed Systems. 2009. **24**, N 3. 205–223.
26. Ament M., Knittel G., Weiskopf D., Strasser W. A parallel preconditioned conjugate gradient solver for the Poisson problem on a multi-GPU platform // Proc. of the 18th EuroMicro Conference on Parallel, Distributed and Network-based Processing. Los Amigos: IEEE Computer Society, 2010. 583–592.
27. DeLong M.A. SOR as a preconditioner. PhD Thesis. University of Virginia. Charlottesville, 1997.
28. Stam J. Stable fluids // Proc. of the 26th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM Press, 1999. 121–128.
29. Harris M.J. Real-time cloud simulation and rendering. PhD Thesis. University of North Carolina. Chapel Hill, 2003.
30. Четверушкин Б.Н. Прикладная математика и проблемы использования высокопроизводительных вычислительных систем // Тр. МФТИ. 2011. **3**, № 4. 55–67.
31. Li W., Wei X., Kaufman A. Implementing lattice Boltzmann computation on graphics hardware // The Visual Computer. 2003. **19**, N 7-8. 444–456.
32. Rossinelli D., Bergdorf M., Cottet G.-H., Koumoutsakos P. GPU accelerated simulations of bluff body flows using vortex particle methods // J. of Computational Physics. 2010. **229**, N 9. 3316–3333.
33. Wei X., Zhao Y., Fan Z., Li W., Qui F., Yoakum-Stover S., Kaufman A.E. Lattice-based flow field modeling // IEEE Trans. on Visualization and Computer Graphics. 2004. **10**, N 6. 719–729.
34. Tölke J., Krafczyk M. TeraFLOP computing on a desktop PC with GPUs for 3D CFD // Int. J. of Computational Fluid Dynamics. 2008. **22**, N 7. 443–456.
35. Chorin A.J. Numerical solution of the Navier-Stokes equations // Mathematics of Computation. 1968. **22**, N 104. 745–762.
36. Kim J., Moin P. Application of a fractional-step method to incompressible Navier–Stokes equations // J. of Computational Physics. 1985. **59**, N 2. 308–323.
37. Brown D.L., Cortez R., Minion M.L. Accurate projection methods for the incompressible Navier–Stokes equations // J. of Computational Physics. 2001. **168**, N 2. 464–499.
38. Morinishi Y., Lund T.S., Vasilyev O.V., Moin P. Fully conservative higher order finite difference schemes for incompressible flow // J. of Computational Physics. 1998. **143**, N 1. 90–124.
39. Ol’shanskii M.A., Staroverov V.M. On simulation of outflow boundary conditions in finite difference calculations for incompressible fluid // Int. J. for Numerical Methods in Fluids. 2000. **33**, N 4. 499–534.
40. Kirkpatrick M.P., Armfield S.W., Kent J.H. A representation of curved boundaries for the solution of the Navier–Stokes equations on a staggered three-dimensional Cartesian grid // J. of Computational Physics. 2003. **184**, N 1. 1–36.
41. Ingram D.M., Causon D.M., Mingham C.G. Developments in Cartesian cut cell methods // Mathematics and Computers in Simulation. 2003. **61**, N 3-6. 561–572.
42. Li Z., Lai M.-C. The immersed interface method for the Navier–Stokes equations with singular forces // J. of Computational Physics. 2001. **171**, N 2. 822–842.
43. Ghias R., Mittal R., Dong H. A sharp interface immersed boundary method for compressible viscous flows // J. of Computational Physics. 2007. **225**, N 1. 528–553.
44. Lee J., Kim J., Choi H., Yang K.-S. Sources of spurious force oscillations from an immersed boundary method for moving-body problems // J. of Computational Physics. 2011. **230**, N 7. 2677–2695.
45. Hung Seo J., Mittal R. A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations // J. of Computational Physics. 2011. **230**, N 19. 7347–7363.
46. Roma A.M., Peskin C.S., Berger M.J. An adaptive version of the immersed boundary method // J. of Computational Physics. 1999. **153**, N 2. 509–534.
47. Pourquie M., Breugem W.P., Boersma B.J. Some issues related to the use of immersed boundary methods to represent square obstacles // Int. J. for Multiscale Computational Engineering. 2009. **7**, N 6. 509–522.
48. Domenichini F. On the consistency of the direct forcing method in the fractional step solution of the Navier–Stokes equations // J. of Computational Physics. 2008. **227**, N 12. 6372–6384.
49. Guy R.D., Hartenstine D.A. On the accuracy of direct forcing immersed boundary methods with projection methods // J. of Computational Physics. 2010. **229**, N 7. 2479–2496.
50. Taira K., Colonius T. The immersed boundary method: a projection approach // J. of Computational Physics. 2007. **225**, N 2. 2118–2137.
51. Mori Y., Peskin C.S. Implicit second-order immersed boundary methods with boundary-mass // Computer Methods in Applied Mechanics and Engineering. 2008. **197**, N 25-28. 2049–2067.
52. Pinelli A., Naqavi I.Z., Piomelli U., Favier J. Immersed-boundary methods for general finite-difference and finite-volume Navier–Stokes solvers // J. of Computational Physics. 2010. **229**, N 24. 9073–9091.

53. Винников В.В., Ревизников Д.Л. Применение декартовых сеток для решения уравнений Навье–Стокса в областях с криволинейными границами // Математическое моделирование. 2005. **17**, № 8. 15–30.
54. Мортиков Е.В. Применение метода погруженной границы для решения системы уравнений Навье–Стокса в областях сложной конфигурации // Вычислительные методы и программирование. 2010. **11**, № 1. 36–46.
55. Gao T., Tseng Y.-H., Lu X.-Y. An improved hybrid Cartesian/immersed boundary method for fluid-solid flows // Int. J. for Numerical Methods in Fluids. 2007. **55**, N 12. 1189–1211.
56. Udaykumar H.S., Mittal R., Rampunggoon P., Khanna A. A sharp interface Cartesian grid method for simulating flows with complex moving boundaries // J. of Computational Physics. 2001. **174**, N 1. 345–380.
57. Braza M., Chassaing P., Minh H.H. Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder // J. of Fluid Mechanics. 1986. **165**. 79–130.
58. Liu C., Zheng X., Sung C.H. Preconditioned multigrid methods for unsteady incompressible flows // J. of Computational Physics. 1998. **139**, N 1. 35–57.
59. Calhoun D. A Cartesian grid method for solving the two-dimensional streamfunction–vorticity equations in irregular regions // J. of Computational Physics. 2002. **176**, N 2. 231–275.
60. Herfjord K. A study of two-dimensional separated flow by a combination of the finite element method and Navier–Stokes equations. Dr. Ing. Thesis. University of Trondheim. Trondheim, 1996.
61. Berthelsen P.A., Faltinsen O.M. A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries // J. of Computational Physics. 2008. **227**, N 9. 4354–4397.
62. Wu Y.L., Shu C. Application of local DFD method to simulate unsteady flows around an oscillating circular cylinder // Int. J. for Numerical Methods in Fluids. 2008. **58**, N 11. 1223–1236.
63. Williamson C.H.K. Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low Reynolds numbers // J. of Fluid Mechanics. 1989. **206**. 579–627.
64. Schäfer M., Turek S. The benchmark problem flow around a cylinder // Flow Simulation with High-Performance Computers II. Notes on Numerical Fluid Mechanics. Hirschel (ed.). **52**. Vieweg: Wiesbaden, 1996. 547–566.
65. Ongoren A., Rockwell D. Flow structure from an oscillating cylinder. Part 1. Mechanisms of phase shift and recovery in the near wake // J. of Fluid Mechanics. 1988. **191**. 197–223.
66. Bishop R.E.D., Hassan A.Y. The lift and drag forces on a circular cylinder oscillating in a flowing fluid // Proc. of the Royal Society of London. Series A. Mathematical and Physical Sciences. 1964. **277**, N 1368. 51–75.
67. Guilmineau E., Queutey P. A numerical simulation of vortex shedding from an oscillating circular cylinder // J. of Fluids and Structures. 2002. **16**, N 6. 773–794.
68. Глазунов А.В. Вихреразрешающее моделирование турбулентности с использованием смешанного динамического локализованного замыкания. Часть I. Формулировка задачи, описание модели и диагностические численные тесты // Изв. РАН. Физика атмосферы и океана. 2009. **45**, № 1. 7–28.
69. Sagaut P. Large eddy simulation for incompressible flows. An introduction. Berlin: Springer, 2006.

Поступила в редакцию
30.01.2012
